

# **Modelling Framework of a Traceability System to Improve Knowledge Sharing and Collaborative Design**

**Farouk Belkadi, Eric Bonjour, Maryvonne Dulmet**

Laboratoire d'Automatique de Besançon  
UMR 6596 - CNRS - ENSMM - UFC  
24, Rue A. Savary - 25000 Besançon (France)  
Tél. (33) 381 402 811 - Fax. 33 381 402 809  
{fbelkadi ; ebonjour ; mdulmet} @ens2m.fr

**Abstract.** In collaborative design, the results of each activity imply modifications of different objects of the situation and are likely to affect the achievement of other activities. The use of collaborative tools enhances the capitalization process, especially at the stage of information collecting. Conversely, capitalized knowledge can also promote cooperation between actors regarding their situation in common. This paper develops a new modelling framework of a traceability system, based on the concept of work situation, to improve knowledge sharing and collaborative design. It aims at giving designers a collaborative tool to capture information of their work and, simultaneously, a view of the progress of their activity and of other inter-related activities.

## **1. Introduction**

Design activities are very complex because designers have to take into account many interactions between various parameters (human, technological, decisional, organizational, etc.) and because they manipulate and share a large amount of knowledge. Complex artefacts, such as cars, are defined by the interactions of numerous actors, working on different elements of the product in question. Moreover, New Products Development (NPD) projects are limited by having less and less time allotted to them. In such a competitive context, collaborative processes, knowledge sharing and the use of collaborative tools among many interdependent actors are keys to a NPD project's success [1], [2]. An efficient way to share knowledge is to use communication and information facilities, especially the dedicated design tools and systems [3]. Because of their use every day, these tools can make the real time capitalization process easier and promote interactions between actors. At the same time, every cooperative actor can obtain plenty of information about the working environment in which he interacts. Stored information is also used to recognize the evolution of activities in the global process (new intermediary results, new constraints, new state of resources, etc.). The actor identifies the requirements of his partners from the present situation and the effects of their actions on his own design situation.

The main idea of our work is based on the contextual and relational character of knowledge building and sharing in collaborative design. We assume that there are strong relationships between collaborative work and knowledge capitalization and re-use. As people construct knowledge when they interact in a social context [4] (work situation), they need to share common knowledge to achieve the coherence and the performance of their collaborative tasks.

On the other hand, traceability, that is, the information acquisition about the progress of the collaborative work (recording of decisions, rules, computations, ...) is a major stage for the building of shared knowledge.

The purpose of this paper is to present a new approach to support traceability and knowledge sharing in collaborative design. This approach focuses on the concepts of work situation and inter-related entities. The workgroup can be considered as a system, composed of various entities, which are involved in different interactions. We intend to use this tool to promote cooperation, and especially to help the designer with his task of traceability at the same time as he is performing his day-to-day activities.

First, in the literature review part, we define the concepts of traceability and collaborative design. Then, we present specifications and a modelling framework of a system for collaborative design oriented towards traceability needs. Finally, a short discussion compares our proposition with other existing CSCW models.

## 2. Literature review

### 2.1. About traceability process in design

In design, traceability is usually associated with project memory, which captures project histories that can subsequently be retrieved and applied to current problems [5]. A modelling framework of a traceability system is used to structure and capture the history of design [6] which concerns relevant information, about the product, its different evolutions, activities, resources, tools, events, and the organization of these elements during the project progress. Various methods, generally associated to KM systems [7], have been developed to meet the need for traceability [8], [9]. The information acquisition stage in these methods is frequently based on analysis of data and on interviews of experts. Different points of view on the design process might be obtained depending on the desired re-usability of the capitalized knowledge. In this sense, it would be useful if designers could be encouraged to use the collaborative tools available to keep a systematic record of the details of their activities and their corresponding situation. We consider that this approach would mean that a designer's work could be kept track of more reliably, and it would contribute to reducing the incidence of forgetting.

## 2.2. About collaborative design

Design is defined in literature as a process of problem solving. Each individual actor builds his own representation of the problem by taking into account the dimensions that are relevant to his intentions. However, shared representations (related to shared global intentions) are needed to ensure coherent integration of individual local results in the collective global actor. The following points insist on two major features of design activity. First, the problems have not yet been completely defined at the beginning of a NPD project. During the life of the project, information will be required from its context (tools, partners, ...) to continue to define the problem, to build a solution and to reduce uncertainty. Second, the design activity is based on an action plan that is constructed progressively during the activity, while at the same time the other actors define their respective requirements according to their local solutions and their own action plans.

A consequence of these points is that design activities are never individual but collective. Collaborative tools are very important in enhancing the design process. In Ergonomics, collective activity is considered as an activity in which a set of people works towards the same goal, consulting each actor, coordinating and cooperating with them. Nabuco [3] distinguishes three kinds of interactions in a workgroup: communication, cooperation and coordination. However, an actor can play many different communication roles during collaboration in a design situation regarding different forms of organization [10]. Due to these reasons, collaborative tools are very important in enhancing the design process [11]. Many collaborative systems called CSCW have been proposed with the aim of assisting actors in their design activity [12], [13]. In [14], a comparative study of some conceptual models of CSCW systems is presented. However, these kinds of systems can be used to help share knowledge during the design process.

The main purpose of our approach is to specify a model of CSCW system to help traceability and also to favour collaboration by sharing context information. The next section develops the specifications for a traceability system.

## 3. Specifications for a traceability system in collaborative design

The main specifications for a traceability system related to collaborative design activities can be formulated as follows:

- S1: To capture and to keep track of the design situation (intermediate results, decision rules, resources, ...)
- S2: To structure and to store relevant information about the evolution of each actor's situation (local situation) and about his contribution to different interactions.
- S3: To share stored information, to enhance each actor's representation about his design situation, i.e., to display relevant views concerning either the global situation or a specific entity.

This system should enhance cooperation by sharing common situation information and by making the detection of new events easier. The main specifications of a

traceability system are formalized in a uses case diagram (fig. 1). On the one hand, each actor should be guided when modifications occur in his work situation. He should obtain information about all the entities that interest him. He should display several views of the situation (for instance, evolution of the constraints generated) and he should be able to ask the other actors for information. On the other hand, every actor acts on the modifications of the situation: he creates new entities and modifies others. He should be able to record these modifications, to share them with others, and therefore to allow traceability of his work.

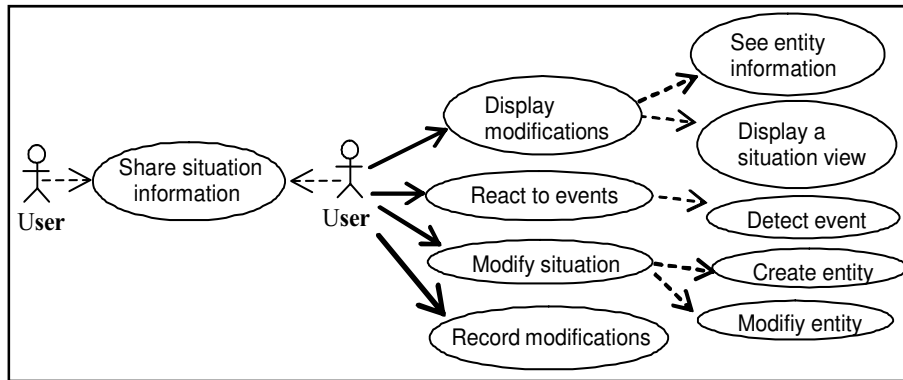


Fig. 1. Uses cases diagram

To meet these specifications, we propose a modelling framework with an appropriate definition of the key-concepts, i.e., situation, entity, specific role.

## 4. Concepts definitions

### 4.1. Situation

In previous publications [15], we discussed the characteristics of the concept of the situation and how to integrate them in order to analyse human action and the contribution made by each actor to any interaction. We defined the concept of work situation as follows: “situation is a set of various entities and of various interactions globally describing the external environment in which an actor mobilizes his competencies”.

### 4.2. Entities

We distinguish two kinds of entities:

- **Basic entities (BE) or concrete entities.** These include all the human actors (called **individual entity**) and the material resources (called **physical entity**), such as product components, work tools, and communication tools.
- **Interactional entities (IE) or abstract entities.** These refer to links between the entities. The three forms of interactional entities are:

- **Operational interactional entities**, which cover the various tasks that an actor has to perform.
- **Community interactional entities**, which establish a membership link between functioning rules.
- **Transactional interactional entities**, which denote various mechanisms of information exchange between actors during the realization of their collective tasks, particularly cooperation and coordination mechanisms.

#### 4.3. Concept of specific roles

The concept of “specific role” represents an actor's interpretation of the collaborative situation. It refers to a set of specific behaviours [16]. This concept is useful for calculating the contribution made by each entity to an interactional entity. We distinguish five kinds of specific roles. With UML (Unified Modeling Language) [17], we model this concept as a generic class with five sub-classes as follows:

- The “actor” role answers the question “who does what?” It concerns every entity who/which participates directly in the interaction.
- The “customer” role answers the question “For whom?” A customer order always precedes need.
- The “manager” role answers the question “How?” It concerns every entity who/which regulates the functioning of an interaction.
- The “support” role answers the question “With what?” It includes every entity who/which indirectly participates in the interaction or assists in its realization.
- The “object” role answers the question “About what?” It concerns every entity on whom/which the interaction acts.

This formulation can make it easier to focus on different aspects of the situation model and can be used to generate useful views.

## 5. Modelling framework

### 5.1. The situation meta-model

Several works, especially in CSCW, cognitive psychology and knowledge management, have focused on giving formal models for the context [18], [19]. In our approach and according to the definition given above (§ 3.1), the meta-model of the situation framework is described in the UML class diagram [17] (figure 2) by a set of

entities and roles. Any object of the entity class can be related to another object of the interactional entity class according to a specific role, which is described as an instance object of the class role. The class “weighting” contains information about the degree of importance given by an actor to any component of the situation.

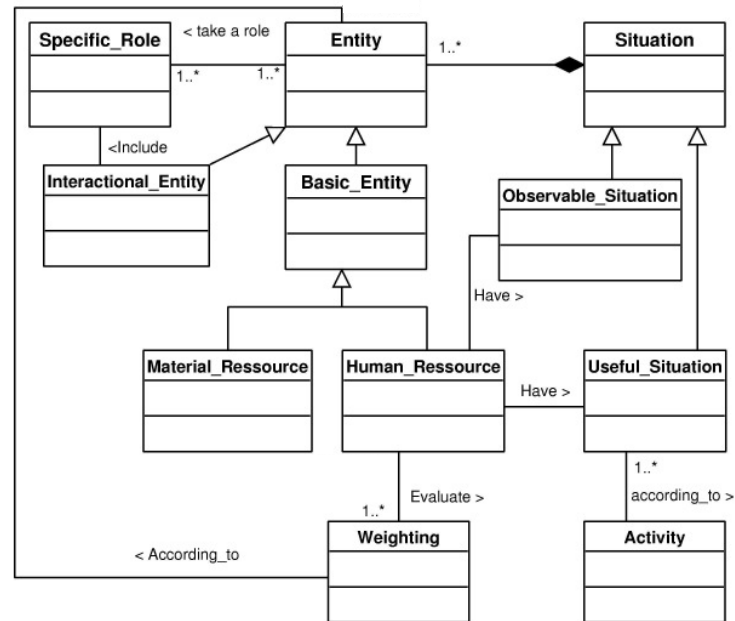


Fig. 2. The situation meta-model

## 5.2. Interaction process

The description of the interaction process is obtained according to the properties of specific roles and their behaviour during the interaction. The UML Activity diagrams are used to model the dynamic aspect of each interaction. For example, the activity diagram (figure 3) illustrates the case of task achieving. The customer expresses his needs through a task (in terms of objectives) and creates the IE "task". This task is allocated by a manager. It follows that a new entity is automatically created: activity (release 0) and the entity that performs the task is an "actor". First, the “actor” analyses and qualifies his situation and then, he defines his action plan (release 0). He may modify this plan structure at any moment of his activity and records the modification in a new release. At the end, the “actor” records the important results of his actions and the activity will be automatically updated in a new release using a new transition. (Consequently, the global situation will be updated). The manager has to observe, follow and regulate the progress of the activity. He has access to the system, and he must define the procedures. He also reports all apparent problems and gives help in solving them.

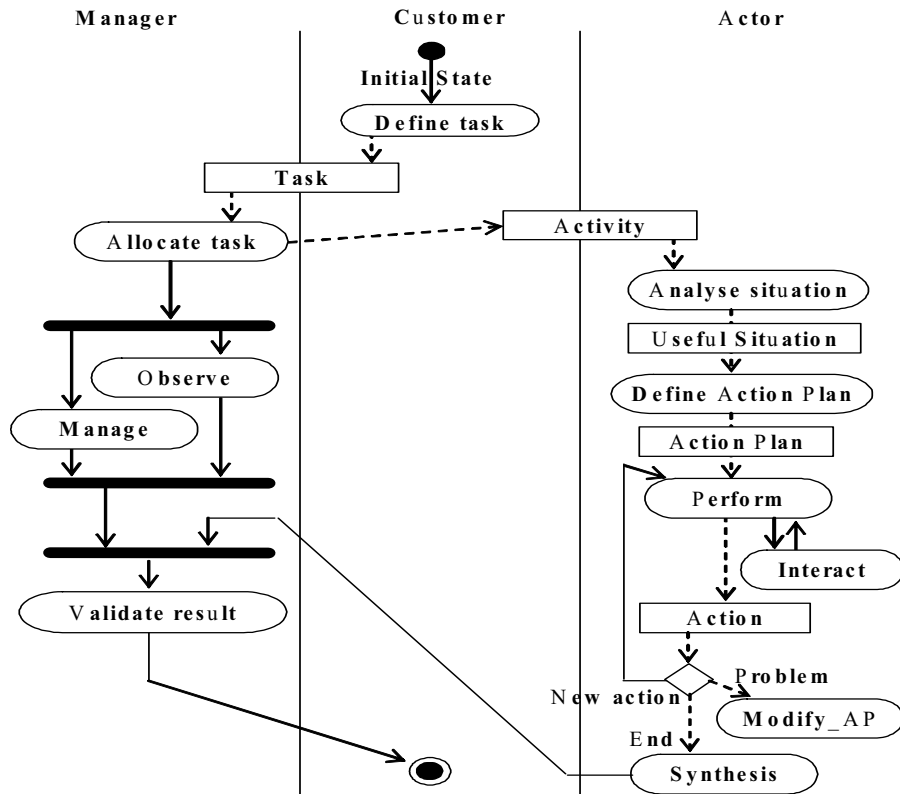


Fig. 3. Interaction process during the task achieving

### 5.3. Sequence diagrams

Each use case or sub-use case is obtained by a set of interactions between the different objects of the system and the user. The UML sequence diagrams are usually used to present the dynamic of these interactions. Figure 4 shows the case of “beginning an activity”. When the actor decides to begin, the system creates and opens a window. Inside the window, the actor defines his action plan and all other important information will be saved in release 0. A new window is opened, the first action is declared: "in progress" and the others "unfinished". The actor systematically records the relevant details of this action and its intermediate results. The "management version" updates the version of the current situation and all its entities and saves the past information in the database. After each validation, which is only relevant for a significant new result, the system updates all the entities involved and shares this new situation release.

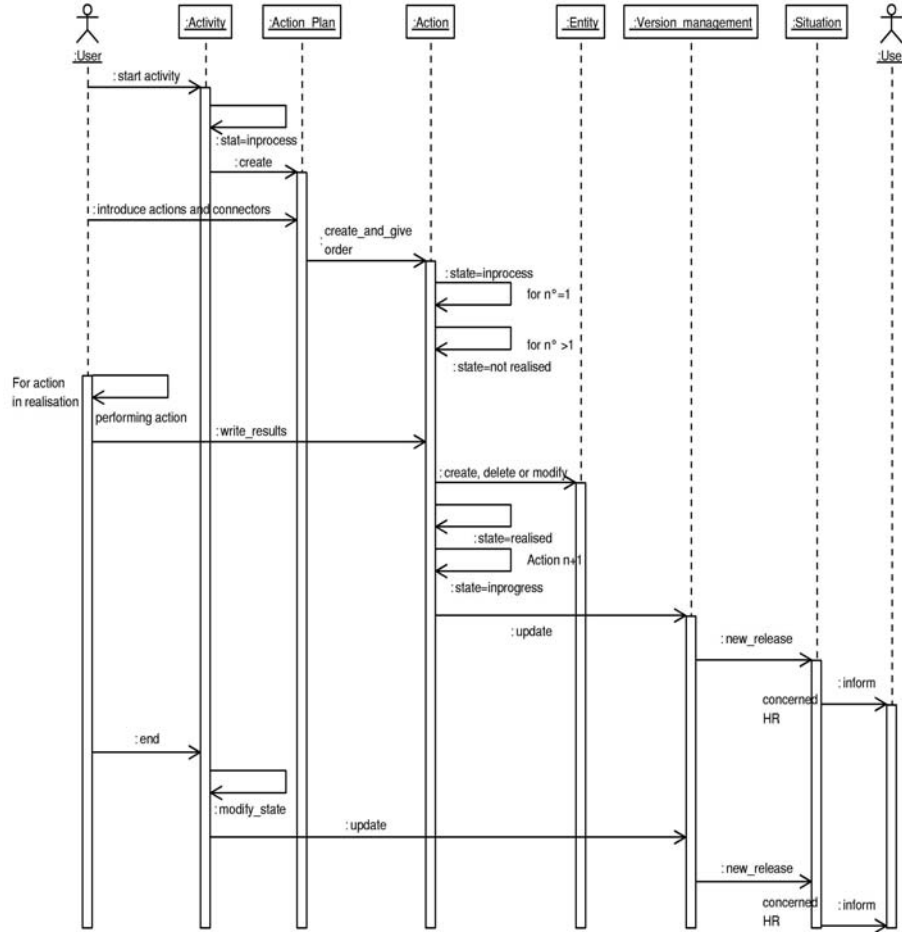


Fig. 4. Activity beginning

## 6. Discussion

In this section, we discuss of the advantages and limits of our approach regarding to other CSCW systems reported in [14].

Generally, each CSCW model aims at covering a particular point of view of the collaborative task. However, four major concepts are frequently used (although sometimes with different names) [14]: activity, actor, resource and tool. The main characteristic of our approach is based on the high abstraction level we used to model work situation. The previous concepts are integrated in our modelling framework but each concept is considered as a sub-class of the class entity.

Similarly with the concept of context in the OoactSM model [20], the situation class in our model is defined as a set of interrelated entities. In our model, Activity is not



associated to a context but it is considered as a part of this one like other kinds of interactions. The concept of interaction (discussed in [14] concerning the Action/Interaction theory) concerns in our model, all collaborative or individual activities but also all relations among the context elements like the constraints, the membership relations (called division of labour in the Activity theory [21]) and different mechanisms of cooperation and coordination between actors.

The second point is that the different models analysed in [14] describe all elements existing in the context but ignore the contribution of any element in the activity and in the interactions. Only the generic model of [14] takes this aspect into account by the attribute role in the class coordination. The model of Task Manager [22] distinguishes two types of roles in collaborative task (participant and observer).

The concept of role is explicitly identified in our approach as an independent object class. A particular interest of the “specific role” class could be expected: to get an original way to manage relevant displays related to the situation information. The aim is to give the user access only to the relevant information with which he is concerned. When the user opens a session, the system searches for all the entities related to him in this situation. According to the user's role in any interactional entity, the system could decide on the level of information to display.

At this moment, little attention is paid to issues related to user interface in our model. For instance, it does not consider the concurrency control dimension used in [14] explicitly, to cope with the simultaneously access to information.

## 7. Conclusion

In this paper, we have proposed a modelling framework of a traceability system to improve knowledge sharing in collaborative design, and then to enhance collaborative activities. This modelling framework could support a new CSCW system in order to allow designers to carry out real-time traceability of their activities and to integrate this task in their day-to-day work. Other benefits can be obtained from this proposal: monitoring the evolution of collective work and facilitating the coordination of it, better understanding of the various mechanisms, which govern collaborative activities. Further research work will present recommendations about the details required for design activities modelling and for the information collecting phase. A prototype is under development and is being tested thanks to an industrial case study.

## References

1. Kvana, T., Candyb L.: Designing collaborative environments for strategic knowledge in design. *Knowledge-Based Systems*, Vol. 13. Elsevier Science (2000) 429-438
2. Liao, S.H.: Knowledge management technologies and applications: literature, review from 1995 to 2002. *Expert Systems with Applications*, Vol. 25. Elsevier Science (2003) 155–164
3. Nabuco, O., Rosário, J. M., Silva, J. R., Drira K.: Scientific Collaboration and Knowledge Sharing in the Virtual Manufacturing Network. INCOM' 2004: 11th IFAC Symposium on INFORMATION CONTROL problems in Manufacturing (2004), Salvador-Bahia, Brazil.

4. Mohrman, S. A., Finegold, D., Mohrman, A.M.: An Empirical Model of the Organization Knowledge System in new product development firms. *Journal of Technology Management*, Vol. 20. Elsevier Science (2003) 7-38
5. Kwan, M., Balasubramanian, P.: KnowledgeScope: managing knowledge in context. *Decision Support Systems*, Vol. 35. Elsevier Science (2003) 467-486
6. Brand, S. C.: A Process Data Warehouse for Tracing and Reuse of Engineering Design process. *The second International Conference On Innovations in Information Technology ICIT'05*, Dubai (2005)
7. Studer, R., Benjamins, V. R., Fensel, D.: Knowledge Engineering: Principles and methods. *Data & Knowledge Engineering*, Vol. 25 (21). Elsevier Science (1998) 161-197
8. Bekhti, S., Matta, N.: A Formal Approach to Model and Reuse the Project Memory. *Journal of Universal Computer Science*, Vol. 6. Springer (2003) 12-22
9. Ramesh, B., Tiwana, A., Mohan, K., Van-Der-Linden, F.: Supporting information product and service families with traceability, *Lecture Notes in Computer Science*, Vol. 2290. Elsevier Science (2002) 353-363
10. Sonnenwald, D. H.: Communication roles that support collaboration during the design process. *Design studies*, Vol. 17. Elsevier Science (1996) 277-301
11. Boujut, JF., Laureillard, P.: A co-operation framework for product-process integration in engineering design. *Design Studies*, Vol. 23. Elsevier Science (2002) 497-513
12. Linfu, S. Weizhi, L: Engineering Knowledge Application in Collaborative Design. *9th International Conference on Computer Supported Cooperative Work in Design*, Coventry, (2005) 722-727
13. Santoyridis, I., Carnduff, T.W., Gray, W.A.: An Object Versioning System to Support Collaborative Design within a Concurrent Engineering Context. *Lecture Notes in Computer Science*, Vol. 1271. *Proceedings of the 15th BNCOD*. Elsevier Science (1997) 184-199
14. Guareis de Farias, C.R., Pires, L.F., Sinderen, M.V.: A Conceptual Model for the Development of CSCW Systems. *4th International Conference on the Design of Cooperative Systems*, France (2000)
15. Belkadi, F., Bonjour, E., Dulmet, M.: Proposition of a Situation Model in View to Improve collaborative design. *11th IFAC Symposium on INformation CONTROL problems in Manufacturing* (2004) Salvador de Bahia, Brazil
16. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology. *The Knowledge Engineering Review*, Vol. 13. M. Uschold (eds.): *Putting Ontologies to Use*. (1998) 1-12
17. Booch, G., Rumbaugh, J., Jacobson, I.: *The Unified Modeling Language: User Guide*. Addison Wesley Longman Publishing Co., Inc. USA (1999)
18. Araujo, R.M., Brézillon, P., Borges, M.R.S., Rosa, M.G.P.: Context Models for managing collaborative software development knowledge. *KI2004 Workshop on Modeling and Retrieval of Context*, Vol. 114. Ulm, Germany (2004)
19. Bernard, A., Hasan, R.: "Working situation" model as the base of life-cycle modelling of socio-technical production systems. *CIRP Design Seminar, Honk-Kong* (2002)
20. Teege, G.: Object-Oriented Activity Support: a Model for Integrated CSCW Systems. *Computer Supported Cooperative Work (CSCW)*, Vol. 5. Springer Science (1996) 93-124
21. Kuutti, K.: The concept of activity as a basic unit of analysis for CSCW research. *Proceeding of 2nd European conference on Computer Supported Cooperative Work (ECSCW'91)* (1991) 249-264
22. Kreifelts, T., Hinrichs, E. and Woetzel, G.: Sharing To-Do Lists with a Distributed Task Manager. *Proceedings of the Third European conference on Computer Supported Cooperative Work (ECSCW'93)* (1993) 31-46